Aerofit

Aerofit is a leading brand in the field of fitness equipment. Aerofit provides a product range including machines such as treadmills, exercise bikes, gym equipment, and fitness accessories to cater to the needs of all categories of people.

Objective

Creating comprehensive customer profiles AeroFit treadmill product through descriptive analysis and Data Visualization. Analayzing data given to reach with the help of two-way contingency tables. Fiding out onditional and marginal probabilities to focus on customer characteristics, enhancing product marketing skills and facilitating improved product recommendations and informed business decisions

Product Portfolio

Aerofit caters to a range of fitness levels with its treadmill offerings:

KP281: An entry-level treadmill priced at USD 1,500.

KP481: A mid-level treadmill for runners, priced at USD 1,750.

KP781: An advanced-feature treadmill priced at USD 2,500.

Dataset Features

The dataset contains the following features:

Product Purchased: Identifies the specific Aerofit treadmill model (KP281, KP481, or KP781) purchased by the customer.

Age: The age of the customer in years.

Gender: The customer's gender (Male/Female).

Education: The number of years of education completed by the customer.

Marital Status: The customer's marital status (Single or Partnered).

Usage: The average number of times per week the customer intends to use the treadmill.

Income: The annual income of the customer (in USD).

Fitness: The customer's self-rated fitness level on a scale of 1 (poor) to 5 (excellent).

Miles: The average number of miles the customer expects to walk/run each week

Importing Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing Dataset

```
data = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749')
```

```
data.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

1.Data analysis steps like checking the structure & characteristics of the dataset

○ The data type of all columns in the “customers” table.

```
data.dtypes
```

```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

○ You can find the number of rows and columns given in the dataset

```
data.shape
```

```
(180, 9)
```

○ Check for the missing values and find the number of missing values in each column

```
data.isnull().sum()
```

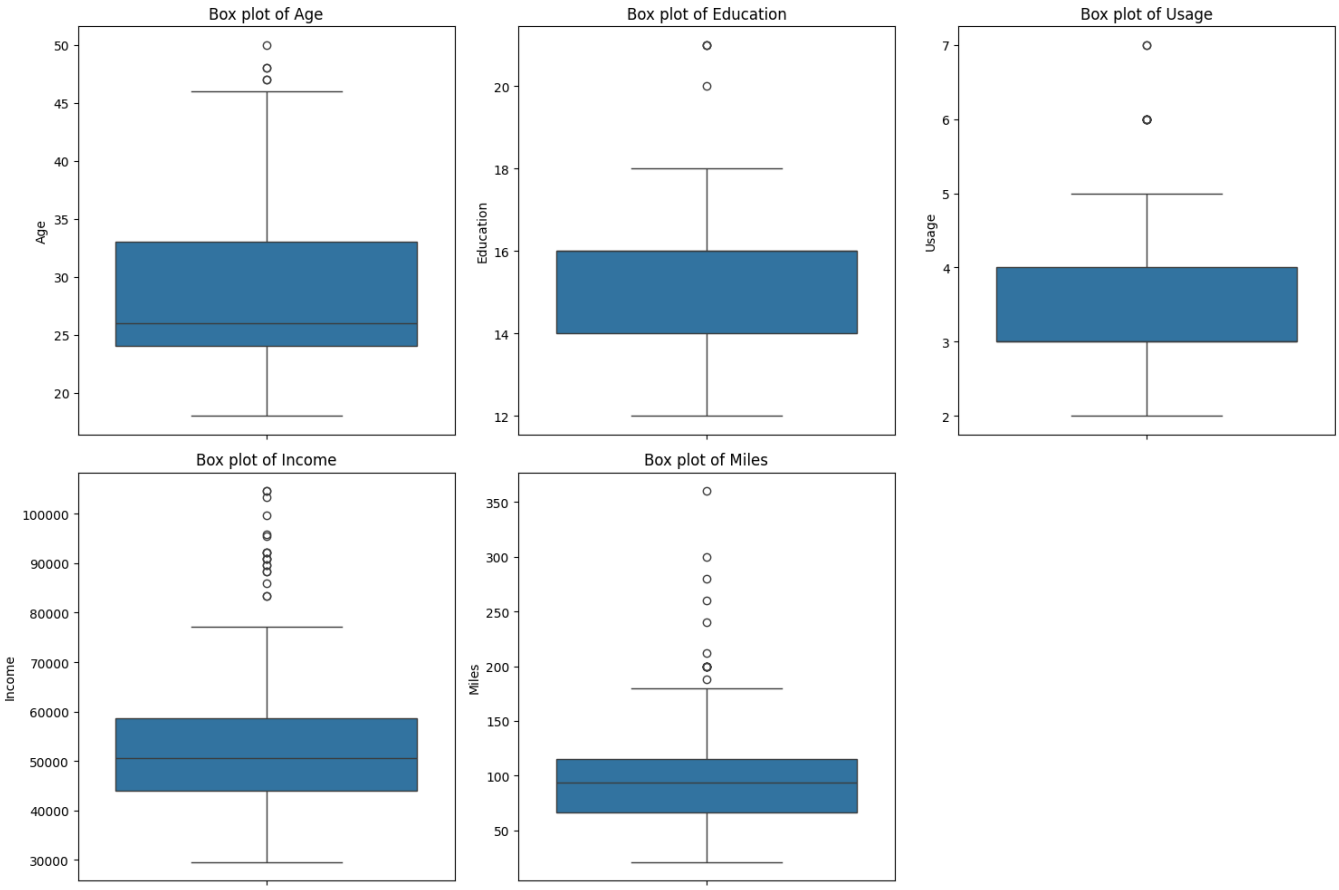
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

2. Detect Outliers

○ Find the outliers for every continuous variable in the dataset

```
continuous_vars = ['Age', 'Education', 'Usage', 'Income', 'Miles']

plt.figure(figsize=(15, 10))
for i, var in enumerate(continuous_vars, 1):
    plt.subplot(2, 3, i)
    sns.boxplot(data=data[var])
    plt.title(f'Box plot of {var}')
plt.tight_layout()
plt.show()
```



○ Remove/clip the data between the 5 percentile and 95 percentile

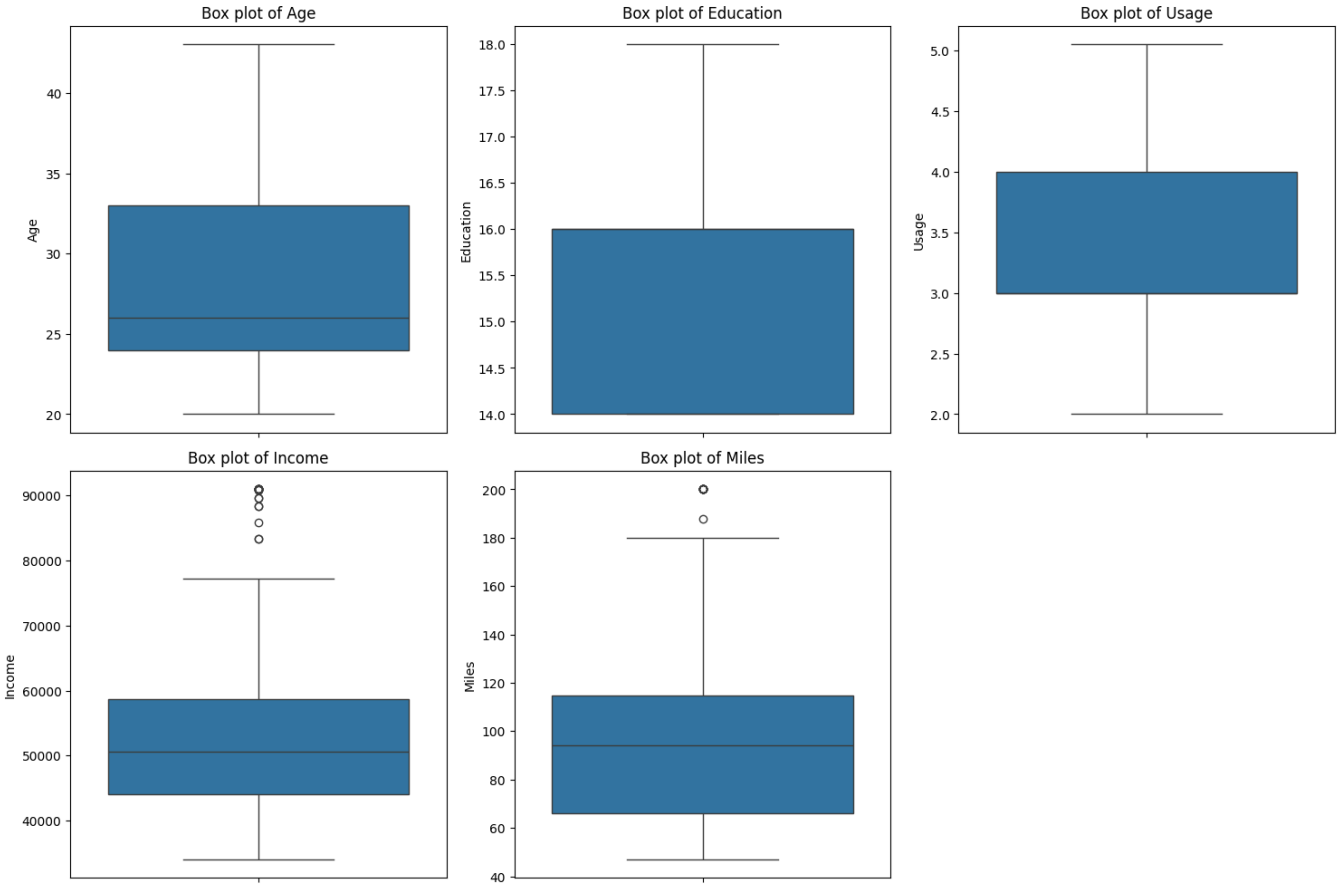
```
def clip_outliers(data, var):
    lower_bound = data[var].quantile(0.05)
    upper_bound = data[var].quantile(0.95)
    data[var] = data[var].clip(lower=lower_bound, upper=upper_bound)
    return data

for var in continuous_vars:
    data = clip_outliers(data, var)


data['Miles'].describe()
```

```
count    180.000000
mean     101.088889
std       43.364286
min       47.000000
25%       66.000000
50%       94.000000
75%      114.750000
max      200.000000
Name: Miles, dtype: float64
```

```
plt.figure(figsize=(15, 10))
for i, var in enumerate(continuous_vars, 1):
    plt.subplot(2, 3, i)
    sns.boxplot(data=data[var])
    plt.title(f'Box plot of {var}')
plt.tight_layout()
plt.show()
```

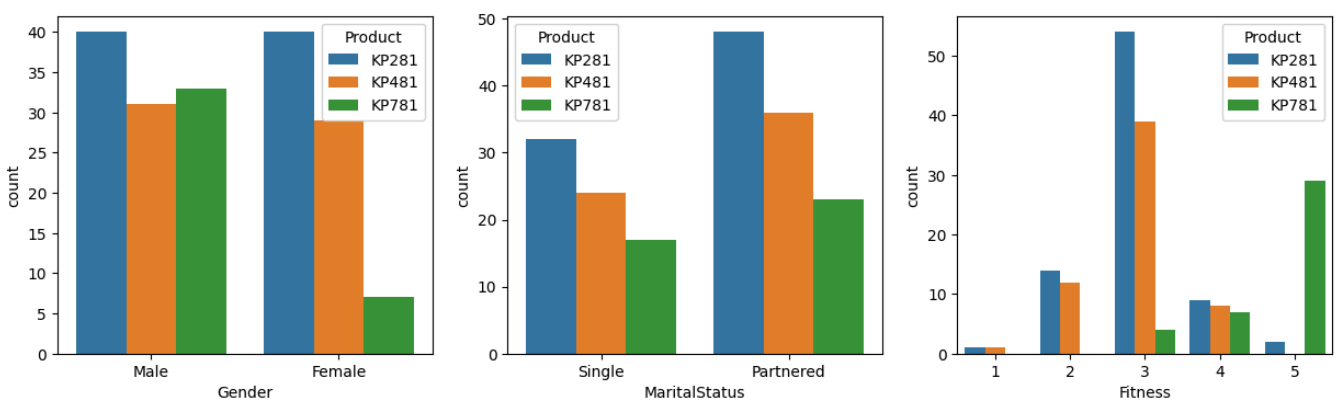


3. Check if features like marital status, Gender, and age have any effect on the product purchased

○ Find if there is any relationship between the categorical variables and the output variable in the data.

Hint: We want you to use the count plot to find the relationship between categorical variables and output variables.

```
plt.figure(figsize = (15,4))
plt.subplot(1,3,1)
sns.countplot(data=data, x='Gender',hue = 'Product')
plt.subplot(1,3,2)
sns.countplot(data=data, x='MaritalStatus',hue = 'Product')
plt.subplot(1,3,3)
sns.countplot(data=data, x='Fitness',hue = 'Product')
plt.show()
```



Usage of products KP281 and KP481 tends to be similar for both Male and female users. Male user tends to use more of product KP781

Partnered users tends to have higher usage than Single users, with KP281 and KP481 as the most used product from the three.

Here we tends to see a pattern as the fitness level increases there is a significant increase in the usage of product KP781.People who consider themselves very fit (ratings 4-5) might be more likely to use KP781

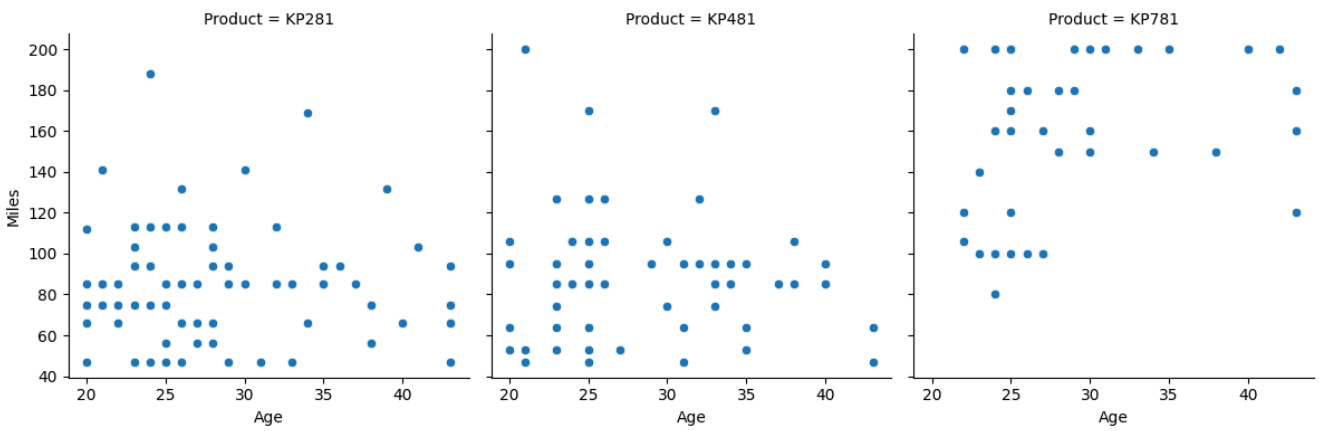
The bars for KP481 and KP281 might be higher for people with lower fitness levels (1-3) compared to those in good or excellent shape (4-5). This suggests these products could be targeted towards beginners or those looking to improve their fitness.

```
data.loc[data['Income'].idxmax()]
```

```
Product              KP781
Age                   28.0
Gender              Female
Education               18
MaritalStatus    Partnered
Usage                 5.05
Fitness                  5
Income             90948.25
Miles                   180
Name: 162, dtype: object
```

○ Find if there is any relationship between the continuous variables and the output variable in the data.

```
g = sns.FacetGrid(data, col="Product", col_wrap=3, height=4)
g.map(sns.scatterplot, "Age", "Miles")
g.add_legend()
plt.show()
```

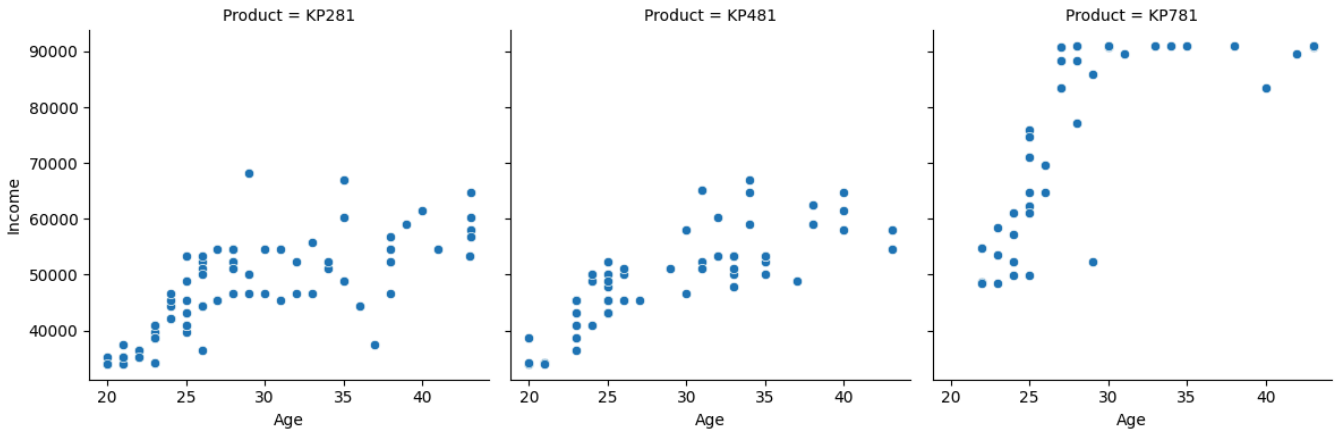


The distribution of miles covered varies among the three products, with KP281 and KP481 showing a wider spread in miles, while KP781 shows a more concentrated range.

Users of all three products fall within a similar age range (20 to 40 years), but their mileage usage differs depending on the product.

KP781 seems to have a more consistent mileage usage among its users compared to KP281 and KP481.

```
g = sns.FacetGrid(data, col="Product", col_wrap=3, height=4)
g.map(sns.scatterplot, "Age", "Income")
#g.add_legend()
plt.show()
```



Positive Correlation between Age and Income

Across all three products, there appears to be a positive correlation between age and income. This means that as people get older, their income tends to increase. This could be due to several factors, such as career advancement, higher wages for experienced workers, or accumulation of wealth over time.

Product KP781 Might Target Higher Income Groups

The data points for product KP781 are generally higher on the y-axis compared to the other two products, suggesting that it might be purchased by people with higher incomes.

Possible Overlap in Target Audiences

There is some overlap between the data series for the three products, indicating that some people in each age group might be interested in all three products.

4. Representing the Probability

○ Find the marginal probability (what percent of customers have purchased KP281, KP481, or KP781)

Hint: We want you to use the pandas crosstab to find the marginal probability of each product.

```
cross_tab = pd.crosstab(index=data['Product'], columns='count')
cross_tab
```

Product	col_0 count
KP281	80
KP481	60
KP781	40

```
marginal_prob = cross_tab / cross_tab.sum() * 100
print('Marginal Probability of each products')
da = pd.DataFrame()
da['Products'] = marginal_prob.index
da['Marginal_Probability'] = marginal_prob.values
da
```

Marginal Probability of each products

	Products	Marginal_Probability
0	KP281	44.444444
1	KP481	33.333333
2	KP781	22.222222

○ Find the probability that the customer buys a product based on each column.

Hint: Based on previous crosstab values you find the probability.

```
prob_by_product = cross_tab /cross_tab.sum() * 100

print("Probability of Buying a Product Based on Each Column:")
print(prob_by_product)
```

```
Probability of Buying a Product Based on Each Column:
col_0        count
Product
KP281    44.444444
KP481    33.333333
KP781    22.222222
```

○ Find the conditional probability that an event occurs given that another event has occurred. (Example: given that a customer is female, what is the probability she'll purchase a KP481)

Hint: Based on previous crosstab values you find the probability.

```
cross = pd.crosstab(data['Product'],data['Gender'])
cross
```

Gender	Female	Male
Product		
KP281	40	40
KP481	29	31
KP781	7	33

```
cond_prob = {}
for prod in cross.index:
  cond_prob[prod]={}
  for val in cross.columns:
    prob_prod_gen = cross.loc[prod,val]
    prob_gen = cross[val].sum()
    cond_prob[prod][val] = prob_prod_gen/prob_gen
print('Conditional Probability\n')
for prod,gend in cond_prob.items():
  for gender,prob in gend.items():
    print(f'Customer is {gender} and have bought {prod} : {prob}\n')
```

```
Conditional Probability

Customer is Female and have bought KP281 : 0.5263157894736842

Customer is Male and have bought KP281 : 0.38461538461538464

Customer is Female and have bought KP481 : 0.3815789473684211

Customer is Male and have bought KP481 : 0.2980769230769231

Customer is Female and have bought KP781 : 0.09210526315789473

Customer is Male and have bought KP781 : 0.3173076923076923
```

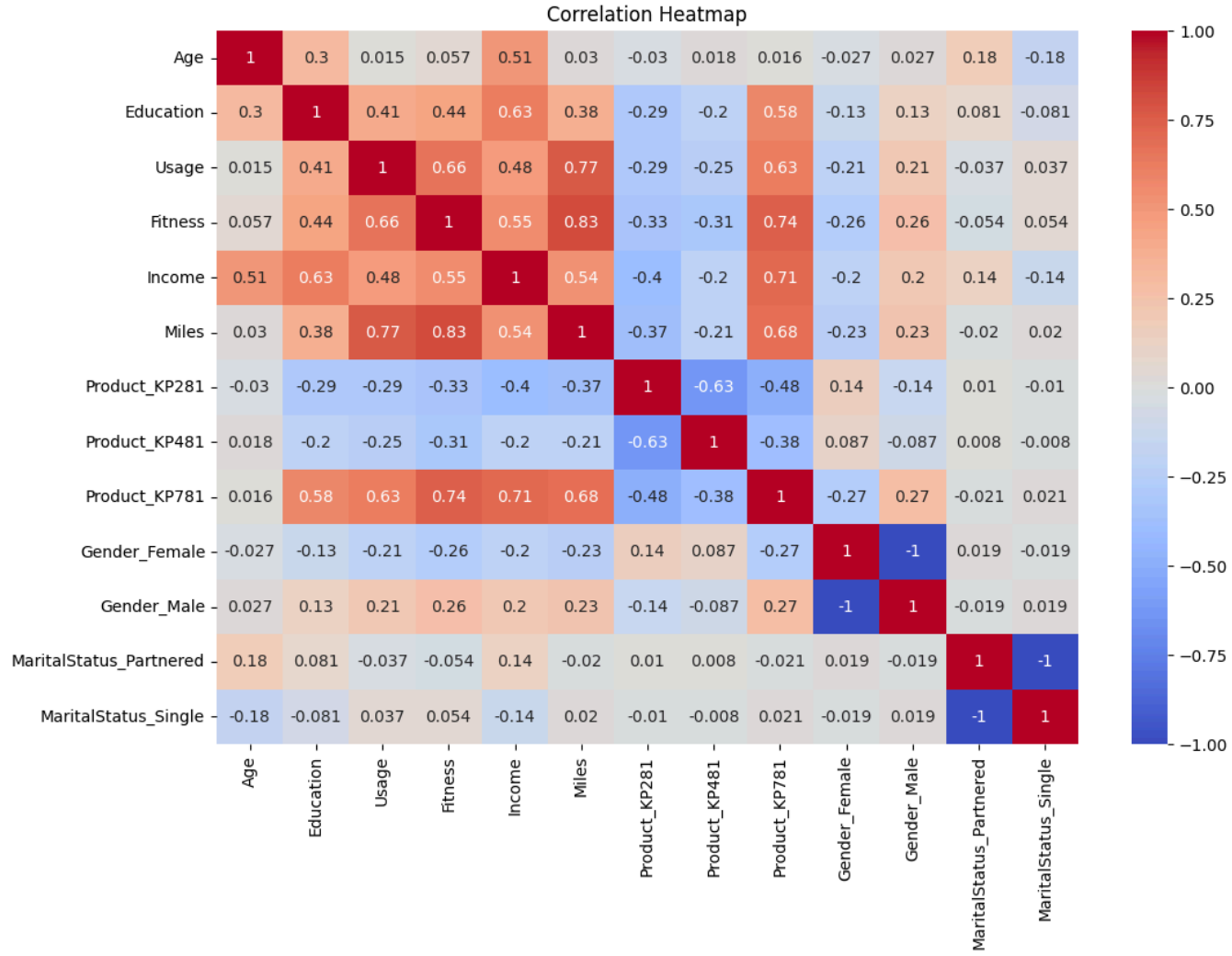
5. Check the correlation among different factors

○ Find the correlation between the given features in the table.

```
df = pd.get_dummies(data, columns=['Product', 'Gender', 'MaritalStatus'])

correlation_matrix = df.corr()


plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Heatmap')
plt.show()
```



6. Customer profiling and recommendation

○ Make customer profilings for each and every product.

Hint: We want you to find at What age, gender, and income group but product the KP281

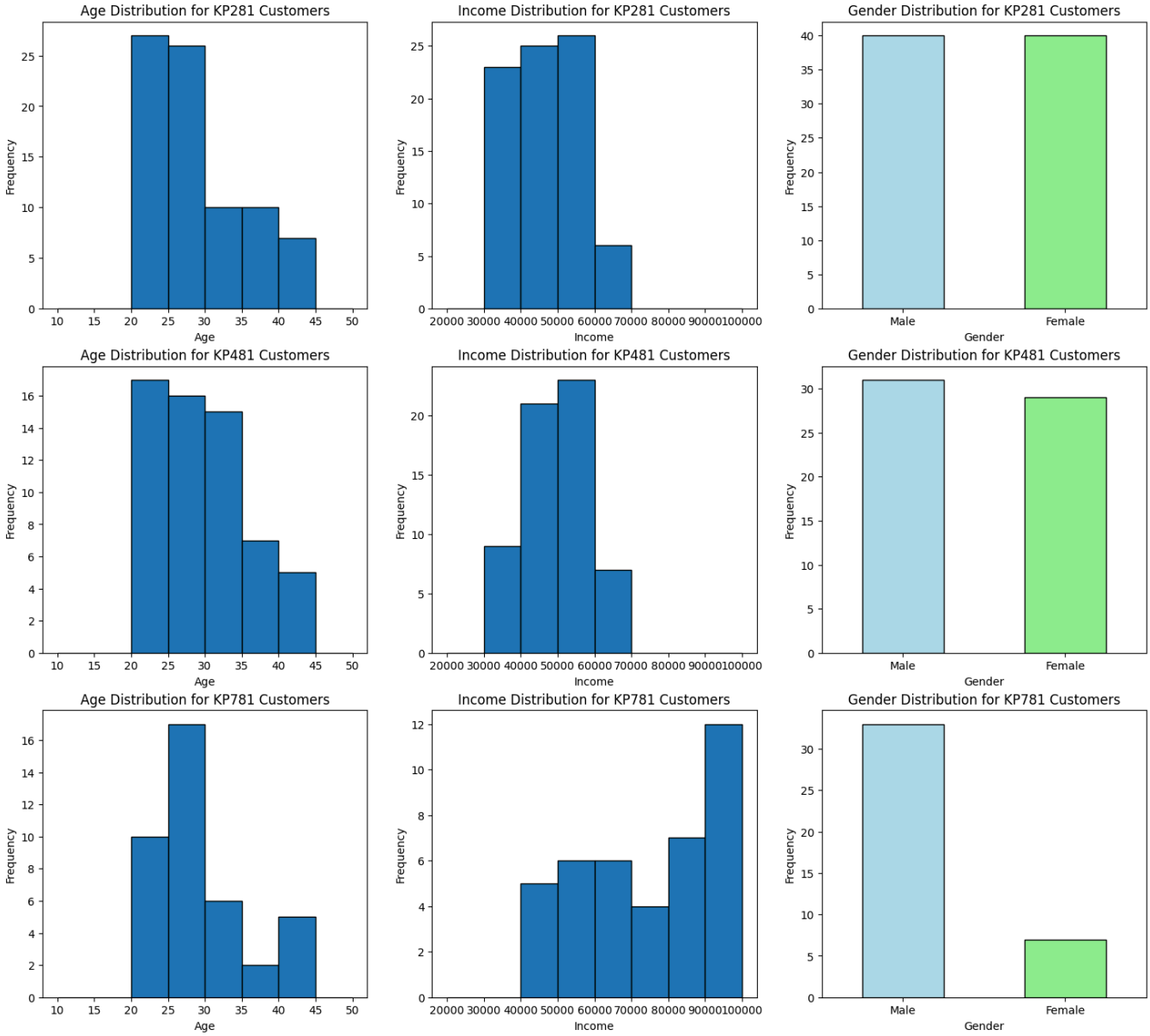
```
dff = data.loc[data['Product'] == 'KP281']
dff1 = data.loc[data['Product']=='KP481']
dff2 = data.loc[data['Product']=='KP781']
```

```
dff2['Income'].describe()
```

```
count       40.000000
mean     73908.281250
std      16572.164368
min      48556.000000
25%      58204.750000
50%      76568.500000
75%      90886.000000
max      90948.250000
Name: Income, dtype: float64
```

```
age_bin = [10,15,20,25,30,35,40,45,50]
income_bin = [20000,30000,40000,50000,60000,70000,80000,90000,100000]
```

```
plt.figure(figsize = (18,16))
plt.subplot(3,3,1)
plt.hist(dff['Age'],bins = age_bin,edgecolor = 'black',align = 'mid')
plt.title('Age Distribution for KP281 Customers')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.subplot(3,3,2)
plt.hist(dff['Income'],bins = income_bin,edgecolor = 'black',align = 'mid')
plt.title('Income Distribution for KP281 Customers')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.subplot(3,3,3)
dff['Gender'].value_counts().plot(kind='bar',color = ['lightblue','lightgreen'],edgecolor = 'black')
plt.title('Gender Distribution for KP281 Customers')
plt.xlabel('Gender')
plt.ylabel('Frequency')
plt.xticks(rotation = 0)
plt.subplot(3,3,4)
plt.hist(dff1['Age'],bins = age_bin,edgecolor = 'black',align = 'mid')
plt.title('Age Distribution for KP481 Customers')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.subplot(3,3,5)
plt.hist(dff1['Income'],bins = income_bin,edgecolor = 'black',align = 'mid')
plt.title('Income Distribution for KP481 Customers')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.subplot(3,3,6)
dff1['Gender'].value_counts().plot(kind='bar',color = ['lightblue','lightgreen'],edgecolor = 'black')
plt.title('Gender Distribution for KP481 Customers')
plt.xlabel('Gender')
plt.ylabel('Frequency')
plt.xticks(rotation = 0)
plt.subplot(3,3,7)
plt.hist(dff2['Age'],bins = age_bin,edgecolor = 'black',align = 'mid')
plt.title('Age Distribution for KP781 Customers')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.subplot(3,3,8)
plt.hist(dff2['Income'],bins = income_bin,edgecolor = 'black',align = 'mid')
plt.title('Income Distribution for KP781 Customers')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.subplot(3,3,9)
dff2['Gender'].value_counts().plot(kind='bar',color = ['lightblue','lightgreen'],edgecolor = 'black')
plt.title('Gender Distribution for KP781 Customers')
plt.xlabel('Gender')
plt.ylabel('Frequency')
plt.xticks(rotation = 0)
plt.show()
```



○ Write a detailed recommendation from the analysis that you have done.

Targeted Marketing:

KP281 and KP481: Focus marketing efforts on younger customers (20-30 years old) with mid-range incomes ($30,000—70,000). Utilize social media platforms popular with this age group, such as Instagram and TikTok.

KP781: Emphasize marketing towards young males. Explore platforms like YouTube and gaming communities where younger males are more active.

Product Bundling:

Create bundles that cater to the income levels and preferences of the primary age group (20-30 years). For instance, offering discounts on bundles that include KP281 and KP481 products might appeal to customers looking for value deals.

Expand Gender-Specific Campaigns:

KP781: Develop campaigns that specifically target young females to balance the gender disparity. Highlight features of KP781 that may appeal more to female customers.

KP281 and KP481: Maintain balanced marketing strategies but include elements that appeal to both genders to retain the current balance.

Loyalty Programs:

Implement loyalty programs that reward frequent purchases, especially targeting the high-concentration age groups (20-30 years old). This could include discounts, early access to new products, or exclusive offers.

Income-Based Promotions:

Offer flexible payment plans or financing options for higher-income customers who may be interested in premium products like KP781. This could increase accessibility and appeal to a broader income range.

Product Exchange Promotions:

Exchange options would provide an additional push to the existing users who are willing to upgrade their product from the lower variant to higher ones, by providing 5-10% discount for their exchange.

Product Feedback:

Collect feedback from the younger demographic to understand their needs and preferences better. Use this information to improve existing products and develop new ones that cater to their lifestyle and expectations.

##